

DILLON & YUDELL LLP
ATTORNEYS AT LAW

**RECEIVED
CENTRAL FAX CENTER**

OCT 05 2005

USPTO FACSIMILE TRANSMITTAL SHEET

TO:	FROM:	
Examiner Rachna Singh	Eustace P. Isidore, Reg. No. 56,104	
ORGANIZATION:	DATE:	
US Patent and Trademark Office	October 5, 2005	
ART UNIT:	CONFIRMATION NO.:	TOTAL NO. OF PAGES INCLUDING COVER:
2176		20
FAX NUMBER:	APPLICATION SERIAL NO.:	
571-273-8300	09/406,435	
ENCLOSED:	ATTORNEY DOCKET NO.:	
Appeal Brief	AT9-99-367	

☒ URGENT ☐ FOR REVIEW ☐ PLEASE COMMENT ☐ PLEASE REPLY ☐ PLEASE RECYCLE

NOTES/COMMENTS:

This fax from the law firm of Dillon & Yudell LLP contains information that is confidential or privileged, or both. This information is intended only for the use of the individual or entity named on this fax cover letter. Any disclosure, copying, distribution or use of this information by any person other than the intended recipient is prohibited. If you have received this fax in error, please notify us by telephone immediately at 512.343.6116 so that we can arrange for the retrieval of the transmitted documents at no cost to you.

8911 N. CAPITAL OF TEXAS HWY., SUITE 2110, AUSTIN, TEXAS 78759
512.343.6116 (V) • 512.343.6446 (F) • DILLONYUDELL.COM

**RECEIVED
CENTRAL FAX CENTER****OCT 05 2005****IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES****ATTORNEY DOCKET NO.: AT9-99-367**In re Application of:
VIKTORS BERSTISSerial No.: **09/406,435**Filed: **SEPTEMBER 27, 1999**For: **METHOD, SYSTEM AND
COMPUTER PROGRAM PRODUCT
FOR KEEPING FILES CURRENT**§
§
§
§
§
§
§
§
§Examiner: **SINGH, RACHNA**Art Unit: **2176****APPEAL BRIEF UNDER 37 C.F.R. 41.37**Mail Stop Appeal Briefs - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

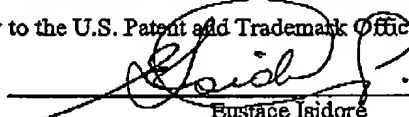
This Brief is submitted in support of the Appeal of the Examiner's final rejection of Claims 1-4, 6-15, 17-26 and 28-33 in the above-identified application. A Notice of Appeal was filed in this case on August 5, 2005 and received in the United States Patent and Trademark Office on August 6, 2005. Please charge the fee of \$500.00 due under 37 C.F.R. §1.17(c) for filing the brief, as well as any additional required fees, to Deposit Account No. 09-0447.

CERTIFICATE OF MAILING OR TRANSMISSION [37 CFR 1.8(A)]

I hereby certify that this correspondence is being:

☐ deposited with the U.S. Postal Service on the date shown below with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.☒ transmitted by facsimile on the date shown below to the U.S. Patent and Trademark Office at (571) 273-8300.10/05/05

Date


Eustace Isidore

10/07/2005 MBINAS 00000009 090447 09406435

01 FC:1402

500.00 DA

Atty. Dkt. No. AT9-99-367

Appeal Brief
Page 1

Serial No.: 09/406,435

REAL PARTY IN INTEREST

The real party in interest in the present Application is International Business Machines Corporation, the Assignee of the present application as evidenced by the Assignment set forth at reel 010282, frame 0412.

RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellant, the Appellant's legal representative, or assignee, which directly affect or would be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF CLAIMS

Claims 1-4, 6-15, 17-26, and 28-33 stand finally rejected by the Examiner as noted in the Final Office Action dated June 2, 2005. Claims 5, 16, and 27 were previously canceled. The rejection of Claims 1-4, 6-15, 17-26 and 28-33 is appealed.

STATUS OF AMENDMENTS

No amendments to the claims have been submitted subsequent to the Final Action from which this Appeal is filed.

SUMMARY OF THE CLAIMED SUBJECT MATTER

Appellant's claimed invention define a method, system and computer program product for keeping current a file downloaded to a client computer system 210 from a connected network computer/server 120/130. Client computer (client) 210 depicted in Fig. 2 and as described at page 9-10 includes a processor (220) and an update manager 230 (see page 10, lines 18-24) executing on the processor that provides the various functions illustrated by Fig. 3 and described within the specification at pages 11-14.

Among these functions is evaluating a downloaded file from a source within the network to determine if a source identifier is present in the downloaded file (see page 11, line 8-16 and block 330 of Fig. 3). The downloaded file is stored at the client 210 with a signature string utilized to find the source identifier within the file and one or more identifying parameters from

among: (1) a locator string identifying a network location from which the file is sourced; (2) a date/time and version number of said file; and (3) a checksum string covering prior entries of said file (*id.*). At page 14, lines 1-4 (and block 350 of Fig. 3), the source is periodically checked to determine if a newer version of the downloaded file exists, and then, as described at page 14, lines 5-33 (and block 360) the older version of the downloaded file is replaced at the client with a complete copy of a newer version when the newer version exists at said source.

Other functions defined by the specification and recited by the claims include, for example, (1) attaching a source identifier to the file when no source identifier is present to indicate the network location from which the file was downloaded (*see* page 11, line 16- page 12, line 24 and Fig. 3, blocks 330, 340), (2) prompting a user to choose when to download a newer version of the file (*see* page 14, lines 8-13) and only downloading the newer version when the user so chooses (*id.* at lines 10-15), and (3) renaming a previous version of the downloaded file to an archived name when a newer version is stored with the current working name (*id.* at lines 13 -17).

GROUND'S OF REJECTION TO BE REVIEWED ON APPEAL

- A. The Examiner's rejection of Claims 1-3, 6, 9-14, 17, 20-25, 28 and 31-33 under 35 U.S.C. §102(e) as being anticipated by *Ball, et al.* (U.S. Patent No. US 2002/0120648) (*hereinafter Ball*) is to be reviewed on Appeal.
- B. The Examiner's rejection of Claims 4, 15 and 26 under 35 U.S.C. §103(a) as being unpatentable over *Ball* is to be reviewed on Appeal.
- C. The Examiner's rejection of Claims 8, 19 and 30 under 35 U.S.C. §103(a) as being unpatentable over *Ball* in view of *Kullick, et al.* (U.S. Patent No. 5,764,992) (*hereinafter Kullick*) is to be reviewed on Appeal.
- D. The Examiner's rejection of Claims 7, 18 and 29 under 35 U.S.C. §103(a) as being unpatentable over *Ball* in view of *Smith, et al.* (U.S. Patent No. 6,006,206) (*hereinafter Smith*) is to be reviewed on Appeal.

ARGUMENT

A. The Examiner's rejection of Claims 1-3, 6, 9-14, 17, 20-25, 28 and 31-33 under 35 U.S.C. §102(e) as being anticipated by *Ball* is not well founded and should be reversed.

Claims 1, 12, and 23

The Examiner has rejected the above claims under 35 U.S.C. 102(e) as being anticipated by *Ball*. *Ball* does not anticipate Appellant's claimed invention because *Ball* fails to teach each element recited by these and other claims. Appellant's independent Claims 1, 12, and 23 provide, in relevant part, the following elements:

evaluating at said client a downloaded file ...a source identifier is present in said downloaded file, ... stored at said client with a name string and a signature string, different from the name string and utilized to find said source identifier within said file and ... a locator string identifying a network location from which the file is sourced; ... and

replacing said downloaded file at said client with a complete copy of said newer version when said newer version of said downloaded file exists at said source.

(emphasis added);

The various deficiencies in *Ball* with respect to the above claim elements are now described. First, *Ball* does not teach a downloaded file stored at the client with a "source identifier," "a signature string utilized to find said source identifier within said file," and/or "a locator string identifying a network location from which the file is sourced." Examiner specifically states that *Ball* teaches "copying an original document ... on a server separate from the WWW" (emphasis added). Examiner then distinguishes that the "server" is different from a "client," by stating that "[i]n response to a request from a client to access a document, a current version ... is presented." What Examiner has failed to recognized, however, is that he has tacitly agreed that the various features and processes described by *Ball* are in fact processes occurring at a server and NOT at a client system.

Ball does not teach a client-level process that includes identifying a file downloaded to a client from a source on a connected network and identifying the source with a specialized string

that is stored with the file. In fact, Ball teaches away from ever having a file stored at the client system.

First, Ball teaches an "EXTERNAL SERVICE" on the network level that maintains a page downloaded from a repository and later receives and stores any changes to the page at the repository. The EXTERNAL SERVICE serves as a mid-level publisher (web site) at which the page may be accessed by various users via their respective client systems. The functionality associated with storage of the page and its updates, etc., as described by *Ball*, occurs solely on the EXTERNAL SERVICE and NOT on the respective client systems of the end-users. This reading of *Ball* is clearly supported by numbered paragraphs 0052, 0053, 0066, 0069, 0071, 0073, 0105-06, among others.

With *Ball*, each user provides a "hot list" of pages (identified by URLs) to the EXTERNAL SERVICE to track changes to the pages specified within the hot list. The EXTERNAL SERVICE periodically checks for updates to the page(s) at the repository and receives these updates dynamically (i.e., with NO user input). Only when the user accesses the page at the EXTERNAL SERVICE via a client system does the updated information get presented to the user (see para. 0076 - 77, 0091, 0127, and 0130).

Second, client-side support is described briefly at para. 0159, which clearly differentiates the client-side feature from the EXTERNAL SERVICE functions. The client features are summarized as a user running a program to "store items in the hotlist locally and run htmdiff against a locally saved copy." *Ball* specifically teaches away from even this limited client-side application, which he describes as "unattractive as the number of pages in the average user's hotlist increases..." No specific mention is provided of storing any network-level parameters such as source identifier along with the downloaded file at the client system, because *Ball* does not teach anything about the client actually downloading and store the file locally. Rather, *Ball* allows the user of the client to pre-select the files via their URLs (entered in the hotlist) and only access the files at the EXTERNAL SERVICE via their URLs, without ever storing the file itself at the local client device.

Third, *Ball* provides a system which enables periodic comparisons of the archived copy of the document at the EXTERNAL SERVICE to the current version at the repository and which updates the archive (EXTERNAL SERVICE) to maintain **“the ability to reconstruct current versions” from the archived copy** (Abstract, etc.). Thus, unlike Appellant’s claimed invention, which specifically downloads a complete copy of the new version of a file from the source to the client, *Ball* retrieves updated portions of the file and sends these updates to a network level EXTERNAL SERVICE. The user of the client may later receive a complete copy of the updated file on the EXTERNAL SERVICE when the user decides to browse to the EXTERNAL SERVICE.

Claims 2, 13 and 24

Claim 2 provides: “attaching, when no source identifier is present, a source identifier to said downloaded file at said client that indicates the network location from which the downloaded file is obtained” (emphasis added). *Ball* does not teach a “source identifier” or adding a “source identifier” to a stored file at the client when the file does not have a source identifier stored therewith. Notably Examiner admits that *Ball* “does not state the term ‘source identifier.’” Examiner then attempts to read into Appellant’s use of that term as being synonymous with a URL. While Appellant’s may utilize a URL as one form of a source identifier, that use is only limited to a specific implementation, at best.

Appellant hereby incorporates by reference the arguments proffered in Amendment E in the present application. Particularly, Appellants would reiterate that as described within Appellant’s specification, which generally provides a local area network (LAN) environment as one environment in which the features of the invention may be applied, files downloaded from LAN-connected servers may often not include source identifiers. Also, at page 11, lines 2-7, possible types of files are described, including a “PDF file” and a “ZIP file,” both of which are traditionally tagged with a file name rather than a source identifier, even when downloaded from the network. Taking a step back from *Ball*, which discusses URLs, provides a clearer picture of what is indeed the focus of Appellant’s invention and distinguishes the client-level response to a general network file update from a PAGE update at a URL that is being tracked at a network-level EXTERNAL SERVICE.

Even assuming, *arguendo*, that Appellant's source identifier is a URL, providing a list of URLs is inherently different from actually adding the URL as an attribute of the file and storing the URL with the file. Further, there is absolutely no teaching within *Ball* of adding a source identifier to an existing file at the client and storing the file along with the source identifier, when one is not previously present.

Examiner incorrectly states that maintaining a list of pages saved is somehow synonymous with or suggestive of adding a source identifier to a stored file that does not currently have a source identifier. Nothing in *Ball* leads to this conclusion, and even Examiner's rejection is at best speculative and not founded on actual teachings within *Ball*. That is Examiner's statement that *Ball* "could attach a source identifier to that page" provides Examiner's opinion of something that *Ball* could have done but which *Ball* does not actually teach. Thus, Examiner's arguments fail to comply with the requirements for a valid § 102 argument. Further, *Ball* would have no reason to attach a source identifier to a stored page when *Ball* specifically downloads the page from a known repository (EXTERNAL SERVICE) using a pre-generated list.

Claims 3, 14, 25

Appellant's Claim 3 recites: "providing an indication to a user that said newer version of said file exists; prompting said user, prior to initiating a download of the newer version, to select ... said newer version; and ... initiating said replacing of said downloaded file by downloading a complete copy of said newer version in place of a present version, wherein when said user does not request said newer version, ... the newer version is not downloaded" (emphasis added). *Ball* does not teach a user selection of when to initiate a download of a newer version of the previously downloaded file. Appellant has reviewed the entire *Ball* reference and found *Ball* completely devoid of any teaching or suggestion of actually prompting a user to select whether to replace the downloaded file with the newer version of the file before the newer version is downloaded to the user's client system, and then initiating the download and update only when the user selects the newer version for download.

Examiner clearly mischaracterizes what is taught by *Ball* when Examiner states that *Ball* teaches that “[i]n response to a request ..., a current version of the document, as archived, is presented.” There is actually no user process within *Ball* directed at updating the versions of the file at the EXTERNAL SERVICE. Notably also, Examiner states that *Ball* teaches “presenting to the user an option to compare selected version as archived ...” (*emphasis added*). *Ball* clearly does not teach prompting the user to select whether to replace the downloaded file at the client system with a new version prior to downloading the new file. First, in *Ball*, the old version of the file is not actually replaced and second, the new version is automatically downloaded and archived within the EXTERNAL SERVICE in order to display to the user the changes between the old version and the new version.

Claims 9, 20, 31

Claim 9 of Appellant’s claims recite: “checking said source responsive to a request to open/access said downloaded file, ... **overriding a current time interval** by initiating said checking step at the time of receipt of the request to open/access said downloaded file and **restarting the current time interval**” (*emphasis added*). *Ball* is devoid of any teaching or suggestion of “overriding a current time interval by initiating said checking step at the time of receipt of the request to open/access said downloaded file and restarting the current time interval” at the client. While *Ball* generally mentions a time period (related to the EXTERNAL SERVICE), there is no mention to or suggestion of overriding the time interval for checking for a newer version whenever the document is opened by the user and restarting the time interval subsequent to the opening. As previously states, no such function is provided at the client as all updates occur at the EXTERNAL SERVICE, which is a separate entity from the “client” referenced by both *Ball* and Appellant’s claims.

Claims 6, 10, 11, 17, 21, 22, 28, 32, 33

All other claims subject to the § 102 rejection depend from Claim 1, 12, and 23 and are therefore covered by the same arguments proffered above rebutting the rejections of the base claims.

The standard for a § 102 rejection requires that the reference teach each element recited in the claims set forth within the invention. For the reasons outlined above, *Ball* fails to meet this standard for all of the various claims. *Ball* therefore does not anticipate Appellant's invention, and the above claims are all allowable. Examiner's § 102 rejection is therefore not well founded and should be reversed.

B. The Examiner's rejection of Claims 4, 15 and 26 under 35 U.S.C. §103(a) as being unpatentable over *Ball* is not well founded and should be reversed.

Claims 4, 15, and 26 each depend on respective independent claims, which have been shown to be allowable over *Ball*. Since these claims depend on allowable claims, they are themselves allowable. Additionally, Appellants note that Examiner merely opines that it would have been obvious to have a URL located in the extended attribute of the downloaded file without linking this conclusory opinion to an actual analysis of the element provided by Appellant's claims against a reference. Appellants disagree with Examiner's unsupported analysis that storing a URL within the extended attribute of a downloaded file would have been obvious. Further, nothing in Appellant's claims specifically states or identifies the source identifier as a URL.

C. The Examiner's rejection of Claims 8, 19 and 30 under 35 U.S.C. §103(a) as being unpatentable over *Ball* in view of *Kullick* is not well founded and should be reversed.

Claims 8, 19, and 30 each depend on respective independent claims, which have been shown to be allowable over *Ball*. Since these claims depend on allowable claims, they are themselves allowable.

D. The Examiner's rejection of Claims 7, 18 and 29 under 35 U.S.C. §103(a) as being unpatentable over *Ball* in view of *Smith* is not well founded and should be reversed.

Claims 7, 18, and 29 each depend on respective independent claims, which have been shown to be allowable over *Ball*. Since these claims depend on allowable claims, they are themselves allowable. Additionally, Appellants note that Examiner references a server-side (or server-implemented) function of "generating and transmitting at a predetermined interval a heartbeat signal" (col. 3, lines 45-53). According to the reference, the heartbeat signal is

transmitted to a client side computer, clearly indicating that the timer is also not at the client computer and not user-adjustable. Further, as provided by the *Smith*, the client site computer receives and processes the heartbeat signal to select "in real time a stale or real-time identifier ... for the formatted real-time financial data based upon the system identifier of the real-time financial data and the heartbeat signal." (id. at line 53-59). This feature is clearly an automatic feature with no user adjustable timer function.

It is clear that one skilled in the art would not have been motivated to combine *Smith* with *Ball* in the manner provided by Examiner. In fact, absent the teaching of Appellant's claimed invention, it is unquestioned that Examiner would not have been inclined to suggest such a combination. Since Examiner may not utilize the teachings of Appellant's specification and claims to find motivation for a § 103 combination, the combination is clearly not valid and thus cannot support a § 103 rejection of the above claims.

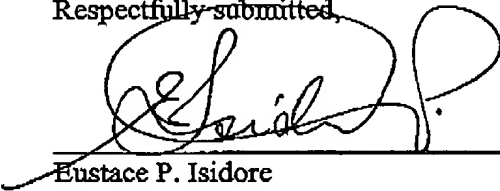
Additionally, Appellant further points out that with respect to Claims 7, 18, and 29, Examiner has clearly mischaracterized what is taught by *Smith* since the cited section of *Smith* fails to teach or suggest a default automatic time interval for initiating a check for updates, where the time interval may be adjusted by the user. That cited section of *Smith* only mentions a "heartbeat signal generator for generating and transmitting at a predetermined interval a heartbeat signal including system identifier," where the heartbeat signal alerts the end devices that the system is alive and providing current updates. Appellant also notes that given that there is no user control of the time for checking for updates within *Ball*, *Ball* would not suggest enabling a user to define a download/update checking interval at the client or later manipulating/overriding that interval.

Given the above reasons, it is clear that neither *Ball* nor any of the above combinations of references suggests key features of Appellant's invention. Thus, one skilled in the art would not find these claims of Appellant's invention unpatentable over *Ball* or the above combinations of references. Examiner's rejections are thus not well founded and should be reversed.

CONCLUSION

Appellant has pointed out with specificity the manifest error in the Examiner's rejections, and the claim language that renders the invention patentable over the combination of references. Appellant, therefore, respectfully requests that this case be remanded to the Examiner with instructions to issue a Notice of Allowance for all pending claims.

Respectfully submitted,



Eustace P. Isidore
Reg. No. 56,104
DILLON & YUDELL LLP
8911 N. Capital of Texas Highway
Suite 2110
Austin, Texas 78759
512-343-6116

ATTORNEY FOR APPELLANT

APPENDIX

1. A method for keeping files current for use in a client computer system coupled to a network, the method comprising the steps of:

evaluating at said client a downloaded file from a source within said network to determine if a source identifier is present in said downloaded file, wherein said downloaded file is stored at said client with a signature string utilized to find said source identifier within said file and one or more identifying parameters from among: (1) a locator string identifying a network location from which the file is sourced; (2) a date/time and version number of said file; and (3) a checksum string covering prior entries of said file;

dynamically checking said source periodically utilizing said source identifier to determine if a newer version of said downloaded file exists; and

replacing said downloaded file at said client with a complete copy of said newer version when said newer version of said downloaded file exists at said source.

2. The method as recited in claim 1 wherein said step of evaluating further includes the step of attaching, when no source identifier is present, a source identifier to said downloaded file at said client that indicates the network location from which the downloaded file is obtained.

3. The method as recited in Claim 1 wherein said step of replacing said downloaded file includes the steps of:

providing an indication to a user that said newer version of said file exists;

prompting said user, prior to initiating a download of the newer version, to select whether to replace said downloaded file with said newer version; and

when said user selects to replace the downloaded file with said newer version, initiating said replacing of said downloaded file by downloading a complete copy of said newer version in place of a present version, wherein when said user does not request said newer version, the present version of said downloaded file on said client is not replaced with the newer version, and the newer version is not downloaded.

4. The method as recited in Claim 1 wherein said source identifier is located in the extended attribute of said downloaded file.
5. (canceled)
6. The method as recited in Claim 1 wherein said source identifier is a uniform resource locator (URL).
7. The method as recited in Claim 1 wherein said step of checking said source periodically includes:
 - defining a default, automatic time interval at which said checking step is initiated; and
 - enabling a user of said client to adjust said time interval, if desired.
8. The method as recited in Claim 1, wherein said replacing step further comprises:
 - renaming a previously stored copy of said downloaded file on said client system from a current working name to an archived name; and
 - storing said newer version of said downloaded file with the current working name of the downloaded file.
9. The method as recited in Claim 1 wherein said step of checking said source comprises checking said source responsive to a request to open/access said downloaded file, wherein, when said checking step is preset to be automatically initiated at a defined periodic time interval, said method further comprises overriding a current time interval by initiating said checking step at the time of receipt of the request to open/access said downloaded file and restarting a current time interval.
10. The method as recited in Claim 1, further comprising storing an identifier and a source descriptor of said downloaded file and each newer version of said downloaded file in a specially coded file registry, which is checked by a controller for correct file location during said checking step.

11. The method as recited in Claim 1 wherein said network is a packet network.
12. A computer system operating in a network environment, comprising:
a processor;
a storage device;
an update manager executing within said processor, including:
means for evaluating a downloaded file from a source within said network to determine if a source identifier is present in said downloaded file, wherein said downloaded file is stored in said storage device with a signature string, different from a name string of the file and utilized to identify said source identifier within said file and one or more identifying parameters from among: (1) a locator string identifying a network location from which the file is sourced; 2 a date/time and version number of said file; and 3 a checksum string covering prior entries of said file;
means for checking said source periodically utilizing said source identifier to determine if a newer version of said downloaded file exists; and
means for replacing said downloaded file at said client with a complete copy of said newer version when the newer version of said downloaded file is present at the source.
13. (currently amended) The computer system as recited in Claim 12 wherein said means for evaluating further includes means for attaching, in response to no source identifier being present, a source identifier to said downloaded file at said client.
14. (currently amended) The computer system as recited in Claim 12 wherein said means for replacing said downloaded file includes:
means for providing an indication to a user that said newer version of said file exists;
means for prompting said user, prior to initiating a download of the newer version, to replace said downloaded file with said newer version; and
means, when said user selects to replace the downloaded file with said newer version, for initiating said replacing of said downloaded file by downloading a complete copy of said newer version in place of a present version, wherein when said user does not request said newer

version, the present version of said downloaded file on said client is not replaced with the newer version, and the newer version is not downloaded.

15. The computer system as recited in Claim 12 wherein said source identifier is located in the extended attribute of said downloaded file.

16. (canceled)

17. The computer system as recited in Claim 12 wherein said source identifier is a uniform resource locator (URL).

18. The computer system as recited in Claim 12 wherein said means for checking said source periodically includes:

means for defining a default, automatic time interval at which said checking step is initiated; and

means for enabling a user to adjust said time interval, if desired.

19. The computer system as recited in Claim 18 wherein said replacing means further comprises:

means for renaming a previously stored copy of said downloaded file on said client system from a current working name to an archived name; and

means for storing said newer version of said downloaded file with the current working name of the downloaded file.

20. The computer system as recited in Claim 12 wherein said means for checking said source comprises checking said source responsive to a request to open/access said downloaded file, wherein, when said checking is preset to be automatically initiated at a defined periodic time interval, said system further comprises means for overriding a current time interval by initiating said checking whenever the request to open/access said downloaded file is received and restarting a current time interval.

21. The computer system as recited in Claim 12, further comprising means for storing an identifier and a source descriptor of said downloaded file and each newer version of said downloaded file in a specially coded file registry, which is checked by a controller for correct file location during said checking step.

22. The computer system as recited in Claim 12 wherein said network is a packet network and said computer system is a client system coupled to said network.

23. A computer program product comprising:

a computer-readable medium having stored thereon computer executable instructions for implementing a method for keeping files current for use in a client computer system coupled to a network, said computer executable instructions when executed, perform the steps of:

evaluating at said client a downloaded file from a source within said network to determine if a source identifier is present in said downloaded file, wherein said downloaded file is stored at said client with a name string and a signature string, different from the name string and utilized to identify said source identifier within said file and one or more identifying parameters from among: (1) a locator string identifying a network location from which the file is sourced; (2) a date/time and version number of said file; and (3) a checksum string covering prior entries of said file;

dynamically checking said source periodically utilizing said source identifier to determine if a newer version of said downloaded file exists; and

replacing said downloaded file at said client with a complete copy of said newer version when said newer version of said downloaded file exists at said source.

24. The computer program product as recited in Claim 23 wherein said step of evaluating further includes the step of attaching, in response to no source identifier being present, a source identifier to said downloaded file at said client.

25. The computer program product as recited in Claim 23 wherein said step of replacing said downloaded file includes the steps of:

providing an indication to a user that said newer version of said file exists;

prompting said user, prior to initiating a download of the newer version, to replace said downloaded file with said newer version; and

when said user selects to replace the downloaded file with said newer version, initiating said replacing of said downloaded file with by downloading a complete copy of said newer version in place of a present version, wherein when said user does not request said newer version, the present version of said downloaded file on said client is not replaced with the newer version, and the newer version is not downloaded

26. The computer program product as recited in Claim 23 wherein said source identifier is located in the extended attribute of said downloaded file.

27. (canceled)

28. The computer program product as recited in Claim 23 wherein said source identifier is a uniform resource locator (URL).

29. The computer program product as recited in Claim 23 wherein said step of checking said source periodically includes:

defining a default, automatic time interval at which said checking step is initiated; and
enabling a user to adjust said time interval, if desired.

30. The computer program product as recited in Claim 29 wherein said replacing step further comprises:

renaming a previously stored copy of said downloaded file on said client system from a current working name to an archived name; and
storing said newer version of said downloaded file with the current working name of the downloaded file.

31. The computer program product as recited in claim 23 wherein said step of checking said URL comprises checking said source responsive to a request to open/access said downloaded file, wherein, when said checking step is preset to be automatically initiated at a defined periodic

time interval, said method further comprises overriding a current time interval by initiating said checking step at the time of receipt of the request to open/access said downloaded file and restarting a current time interval.

32. The computer program product as recited in Claim 23, further comprising storing an identifier and a source descriptor of said downloaded file and each newer version of said downloaded file in a specially coded file registry, which is checked by a controller for correct file location during said checking step.

33. The computer program product as recited in Claim 23 wherein said network is a packet network.

EVIDENCE APPENDIX

Other than the Office Action(s) and reply (or replies) already of record, no additional evidence has been entered by Appellants or the Examiner in the above-identified application which is relevant to this appeal.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings as described by 37 C.F.R. §41.37(c)(1)(x) known to Appellants, Appellants' legal representative, or assignee.